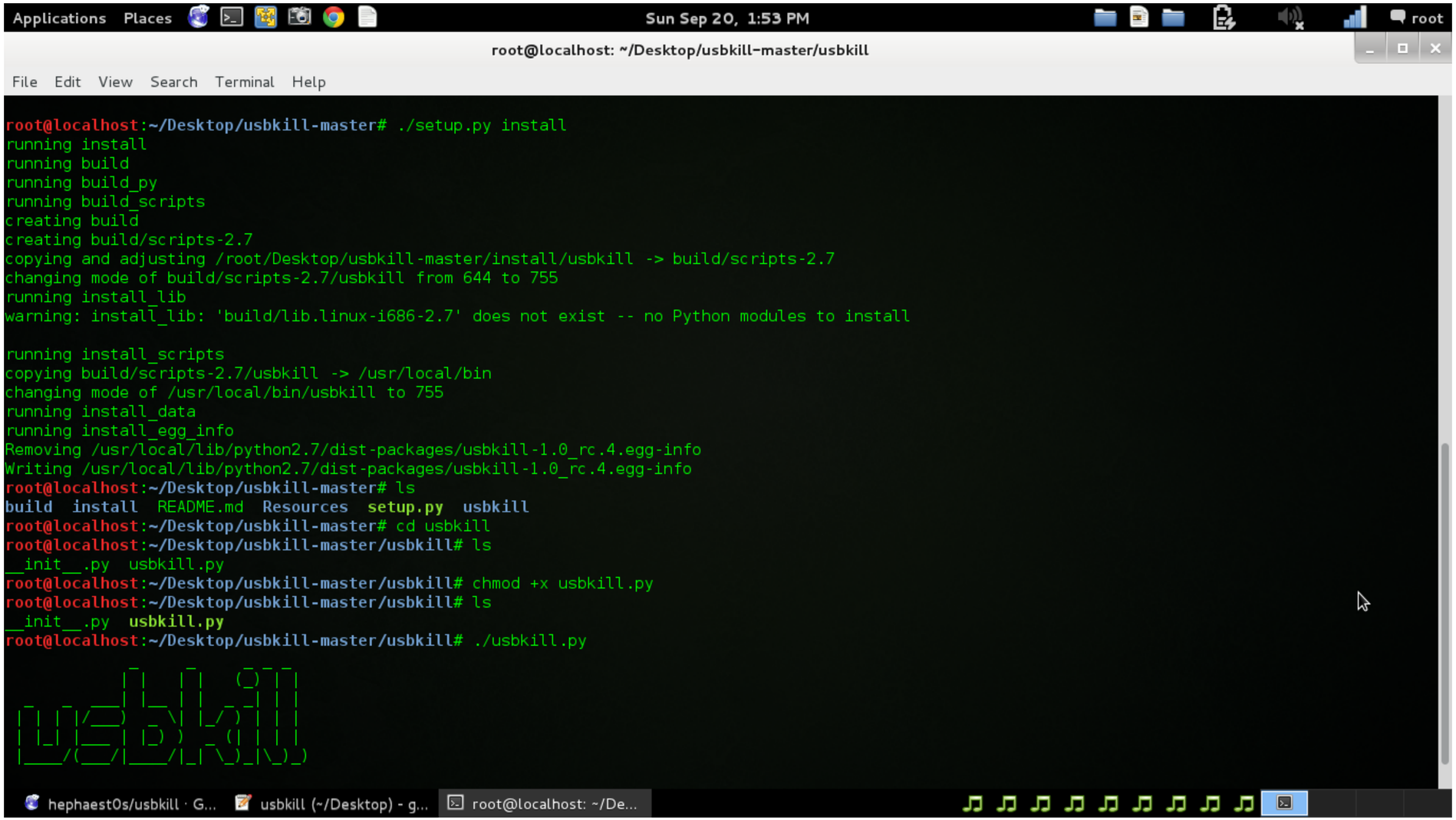
kernKill

Loadable Linux kernel module

@vk_khr_ptr

USB Kill

Not USB Killer



Target Audience

- Activists;
- Hacktivists;
- Journalists;
- Politics;
- Dissidents.

Why Linux

Windows/Mac → GNU/Linux

- Open Source;
- Provable Security;
- Independent from Tech Corporations.

Why Kernel-Space?

User-space:

- User-space service can be disable remotely by any kind of software;
- Easy detectable in systems.

Kernel-space:

- Can’t be unloaded by unprivileged users;
- Can masquerade as other drivers and modules.

Security pre-requirements

- GNU/Linux system (with modern compatible hardware);
- Software encrypted SSD/HDD via dm-crypt;
- Encrypted, updated and locked UEFI;
- Encrypted, updated and locked BootLoader.

Best class solution: most ThinkPad’s

Main Principles

Module architecture

- The module is registered in the kernel as a USB subsystem driver;
  - `.probe = etx_usb_probe(),`
  - `.disconnect = etx_usb_disconnect(),`
- At each probe and disconnect:
  - Check `id->idVendor == USB_VENDOR_ID && id->idProduct == USB_PRODUCT_ID`

```
kill_proc_info(),
kernel_power_off(),
etc.
```

Anti-Forensic Perspective

You’re in trouble, what’s next?

If Security pre-requirements OK that’s enough;

Possible threats:

- Cold boot attacks (get encryption keys from memory) – really rare;

Hackathon Progress

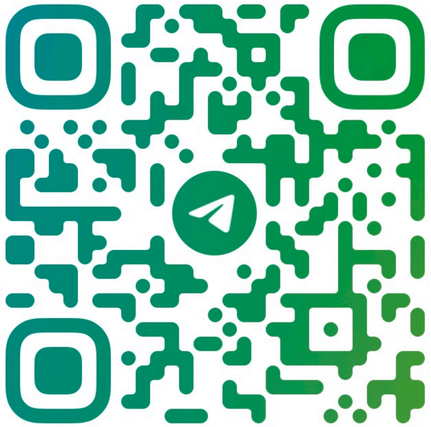
- MVP:
  - USB sub-system registration;
  - `Probe` and `Disconnect` detection;
  - Device verification;
  - Kernel-shutdown and proc-kill.

Plans

- Debug;
- Verification via S/N;
- Experiments with veracrypt containers;
- Create useful configuration tool.

thx

lazy_static!



@VK_KHR_PTR

github